UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/894,205	06/28/2001	Kazushi Kurata	10873.753US01	7105

23552 7590 11/26/2004

MERCHANT & GOULD PC
P.O. BOX 2903
MINNEAPOLIS, MN 55402-0903

EXAMINER	
GERSTL, SHANE F	

ART UNIT	PAPER NUMBER
2183	

DATE MAILED: 11/26/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/894,205	KURATA, KAZUSHI	
	Examiner	Art Unit	
	Shane F Gerstl	2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *01 September 2004*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *25-44* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *25-44* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *28 June 2001* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)☐ Some * c)☐ None of:

1.☒ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.

5) ☐ Notice of Informal Patent Application (PTO-152)

6) ☐ Other: _____.

DETAILED ACTION

1. Claims 25-44 have been examined.

Papers Received

2. Receipt is acknowledged of amendment papers submitted, where the papers have been placed of record in the file.

3. The amendment has successfully overcome the objections to the title, drawings, and claims, all of which are herein withdrawn.

Claim Rejections - 35 USC § 102

4. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

5. Claims 25-44 are rejected under 35 U.S.C. 102(b) as being anticipated by Cocke (3,577,189).

6. In regard to claim 25, Cocke discloses a data processing device reading in and executing instructions in a certain sequence, the data processing device comprising:

a. a fetch portion for reading in a computational instruction; [Figure 1 shows that instruction address are sent to an external storage or instruction memory for fetching as outlined in column 11, lines 65-69. Table III of column 4 shows that two conditional branch instructions (a first and second conditional instruction) are executed along with other operations (OP) and an Exit instruction. Column 1, line 73 – column 2, line 27 show that the conditions of the branches are

determined or computed and thus the conditional branch instructions are computational instructions.]

b. a decoding portion for decoding the computational instruction that has been read in; [Figure 1A, element 40 illustrates an instruction decoder that decodes branch and other instructions as shown in column 2, lines 49-51 and thus the computational instruction is also decoded by this portion.]

c. an execution portion for executing the computational instruction in accordance with a decoding of the computational instruction; [Column 1, line 73 – column 2, line 27 show that instructions are processed (executed) including the branch (computational) instruction. Figure 1A, element 62 shows the function generator that processes or executes the instructions. This execution portion is also shown to receive its data from the instruction decoder and thus the execution is done in accordance with the decoding of the computational instruction.]

d. and an instruction overriding control circuit for controlling an overriding of subsequent instructions that follow the computational instruction in response to a conditional execution status updated by a sequencer in accordance with the decoding of the computational instruction. [Column 4, lines 1-16 show that subsequent branches are ignored or overridden if the condition for a first branch condition is successful (conditional execution status). Column 17, lines 5-65 with further show this to be true with the example of figures 7 and 7A where the computational instruction is branch1. This section also describes the hardware

of figures 1 (including the inhibiting AND gate) used to override the subsequent instructions (and thus the hardware is a instruction overriding control circuit). This section along with figure 1A shows that the status of the condition ("successful" in the example of figures 7) is updated in branch state flip-flop 64. This is done by the result of the conditional function from the function generator and since the hardware involved in figure 1A for updating the flip-flop determines branch conditions and has bearing on the sequence of executed instructions it may be appropriately named a sequencer. As shown in the sections cited above and in Figure 1A, the condition of the branch instruction is executed based on or in accordance with the decoding of the instruction from the instruction decoder 40.]

7. In regard to claim 26, Cocke discloses the data processing device according to claim 25, wherein the instruction overriding control circuit overrides none of the subsequent instructions in response to the conditional execution status updated by the sequencer. [As shown with respect to the example of figure 7 and the associated section of text, if the first branch condition is deemed to be unsuccessful (and thus the branch state flip-flop is updated by the sequencer to reflect this as described above) the subsequent branch is not ignored or overridden but is instead tested as normal.]

8. In regard to claim 27, Cocke discloses a data processing device reading in and executing instructions in a certain sequence, the data processing device comprising:

a. a fetch portion for reading in a computational instruction; [Figure 1 shows that instruction address are sent to an external storage or instruction memory for fetching as outlined in column 11, lines 65-69. Table III of column 4 shows that

two conditional branch instructions (a first and second conditional instruction) are executed along with other operations (OP) and an Exit instruction. Column 1, line 73 – column 2, line 27 show that the conditions of the branches are determined or computed and thus the conditional branch instructions are computational instructions.]

b. a decoding portion for decoding the computational instruction that has been read in; [Figure 1A, element 40 illustrates an instruction decoder that decodes branch and other instructions as shown in column 2, lines 49-51 and thus the computational instruction is also decoded by this portion.]

c. an execution portion for executing the computational instruction in accordance with a decoding of the computational instruction; [Column 1, line 73 – column 2, line 27 show that instructions are processed (executed) including the branch (computational) instruction. Figure 1A, element 62 shows the function generator that processes or executes the instructions. This execution portion is also shown to receive its data from the instruction decoder and thus the execution is done in accordance with the decoding of the computational instruction.]

d. and an instruction overriding control circuit for overriding at least one instruction of subsequent instructions that follow the computational instruction in response to a conditional execution status updated by a sequencer in accordance with the decoding of the computational instruction. [Column 4, lines 1-16 show that subsequent branches are ignored or overridden if the condition

for a first branch condition is successful (conditional execution status). Column 17, lines 5-65 with further show this to be true with the example of figures 7 and 7A where the computational instruction is branch1. This section also describes the hardware of figures 1 (including the inhibiting AND gate) used to override the subsequent instructions (and thus the hardware is a instruction overriding control circuit). This section along with figure 1A shows that the status of the condition ("successful" in the example of figures 7) is updated in branch state flip-flop 64. This is done by the result of the conditional function from the function generator and since the hardware involved in figure 1A for updating the flip-flop determines branch conditions and has bearing on the sequence of executed instructions it may be appropriately named a sequencer. As shown in the sections cited above and in Figure 1A, the condition of the branch instruction is executed based on or in accordance with the decoding of the instruction from the instruction decoder 40.]

9. In regard to claim 28, Cocke discloses the data processing device according to claim 27, wherein the instruction overriding control circuit overrides the at least one instruction by overriding an execution of the at least one instruction in the execution portion. [As shown in the sections cited above, the decoding of an ignored or overridden instruction is halted and thus the execution is also overridden since the instruction cannot execute without having been decoded.]

10. In regard to claim 29, Cocke discloses the data processing device according to claim 27, wherein the instruction overriding control circuit overrides the at least one

instruction in accordance with the conditional execution status and a status flag determined by execution of the computational instruction. [As shown in the sections cited above, the overriding is done based on the status of the branch condition (conditional execution status), which is represented by a status flag in branch status flip-flop 64, based on the execution of the computational instruction (or first branch instruction as in the examples of table 3 and figure 7).]

11. In regard to claim 30, Cocke discloses the data processing device according to claim 27, wherein the computational instruction and a first instruction and a second instruction are sequentially arranged as an instruction sequence, wherein the overriding control circuit overrides either the first instruction or the second instruction in response to the conditional execution status updated by the sequence. [As shown in the example of figure 7, a sequence of instructions includes the computational (branch1), OPα2 (a third instruction), OPα3 (a second instruction), and branch2 (a first instruction) instructions. As shown with respect to this example above, the first instruction (branch2) will be ignored or overridden in response to the conditional execution status of the first branch being updated by the sequencer to a successful state.]

12. In regard to claim 31, Cocke discloses the data processing device according to claim 30, wherein the instruction overriding control circuit overrides either the first instruction or the second instruction in accordance with the conditional execution status and a status flag determined by execution of the computational instruction. [As shown with respect to the example of figure 7 above, the first instruction (branch2) will be ignored or overridden in response to the conditional execution status of the first branch

being updated in the status flag of flip-flop 64 to a successful state based on the execution of the computational instruction.]

13. In regard to claim 32, Cocke discloses the data processing device according to claim 30, wherein neither the first instruction nor the second instruction includes any condition to be overridden. [In the example of figure 7 above, if the condition of computational (branch1) instruction is not successful none of the given subsequent instructions will be overridden.]

14. In regard to claim 33, Cocke discloses the data processing device according to claim 30, wherein the computational instruction and a first instruction group including a plurality of instruction strings and a second group including a plurality of instruction strings are sequentially arranged as an instruction sequence, wherein the overriding control circuit overrides either the first instruction group or the second instruction group in response to the conditional execution status updated by the sequencer. [As shown in the example of figure 7, there is a first instruction group comprising the OPα2 and OPα3 instruction strings and a second group comprising the OPα5 and OPα6 instructions. As shown in column 17, line 5 – column 18, line 11 show that this second group is inhibited or overridden based on the status of the condition of the computational instruction (branch1) being successful, since this prompts the EXIT instruction to start execution of a new routine. As another example, if branch2 was immediately followed by a branch3 instruction, one could identify these two branch instructions to be a second group and as described previously, if the branch1 instruction (computational instruction) has a successful status, this second group would be overridden.]

15. In regard to claim 34, Cocke discloses the data processing device according to claim 33, wherein the instruction overriding control circuit overrides either the first instruction group or the second instruction group in accordance with the conditional execution status and a status flag determined by execution of the computational instruction. [As shown above with respect to the first example for claim 33, the second group is overridden based on the execution of the computational instruction as described. This execution sets a status bit in an inhibit flip-flop 126 of figure 1A as described in column 10, lines 34-53. As shown above for the second example, the second group (branch2 and branch3) will be ignored or overridden in response to the conditional execution status of the first branch being updated in the status flag of flip-flop 64 to a successful state based on the execution of the computational instruction.]

16. In regard to claim 35, Cocke discloses the data processing device according to claim 33, wherein no instructions included in both the first instruction group and the second instruction group include any conditions to be overridden. [The claim language literally means no instructions that exist in both of the groups are overridden based on a condition. With this interpretation, clearly the claim is met since the same instruction does not exist in both of the groups in either example cited above. The Examiner believes Applicant to mean that no instructions in either of the group are overridden based on a condition. With this interpretation, in both the first and second examples above, if the condition of computational (branch1) instruction is not successful none of the given subsequent instructions will be overridden.]

17. In regard to claim 36, Cocke discloses the data processing device according to claim 27, wherein the computational instruction and a first instruction are sequentially arranged as an instruction sequence, wherein the overriding control circuit overrides the first instruction in response to the conditional execution status updated by the sequencer. [The example of figure 7 shows the computational instruction (branch1) and a branch2 instruction that is sequentially arranged (in program order after) as an instruction sequence. As shown with respect to this example above, the first instruction (branch2) will be ignored or overridden in response to the conditional execution status of the first branch being updated by the sequencer to a successful state.]

18. In regard to claim 37, Cocke discloses the data processing device according to claim 36, wherein the instruction overriding control circuit overrides the first instruction in accordance with the conditional execution status and a status flag determined by execution of the computational instruction. [As shown with respect to the example of figure 7 above, the first instruction (branch2) will be ignored or overridden in response to the conditional execution status of the first branch being updated in the status flag of flip-flop 64 to a successful state based on the execution of the computational instruction.]

19. In regard to claim 38, Cocke discloses the data processing device according to claim 36, wherein the first instruction does not include any condition to be overridden. [In the example of figure 7 above, if the condition of computational (branch1) instruction is not successful none of the given subsequent instructions will be overridden.]

20. In regard to claim 39, Cocke discloses the data processing device according to claim 27, wherein the computational instruction and a first instruction group including a plurality of instruction strings are sequentially arranged as an instruction sequence, wherein the overriding control circuit overrides the first instruction group in response to the conditional execution status updated by the sequencer. [As shown in the example of figure 7, there is a first instruction group comprising the OPα5 and OPα6 instructions that are sequentially after the computational (branch1) instruction. As shown in column 17, line 5 – column 18, line 11 show that this group is inhibited or overridden based on the status of the condition of the computational instruction (branch1) being successful, since this prompts the EXIT instruction to start execution of a new routine. As another example, if branch2 was immediately followed by a branch3 instruction, one could identify these two branch instructions to be a group sequentially after the computational instruction and as described previously, if the branch1 instruction (computational instruction) has a successful status, this second group would be overridden.]

21. In regard to claim 40, Cocke discloses the data processing device according to claim 39, wherein the instruction overriding control circuit overrides the first instruction group in accordance with the conditional execution status and a status flag determined by execution of the computational instruction. [As shown above with respect to the first example for claim 33, the group is overridden based on the execution of the computational instruction as described. This execution sets a status bit in an inhibit flip-flop 126 of figure 1A as described in column 10, lines 34-53. As shown above for the second example, the group (branch2 and branch3) will be ignored or overridden in

response to the conditional execution status of the first branch being updated in the status flag of flip-flop 64 to a successful state based on the execution of the computational instruction.]

22. In regard to claim 41, Cocke discloses the data processing device according to claim 39, wherein no instructions included in the first instruction group include any conditions to be overridden. [In both the first and second examples above, if the condition of computational (branch1) instruction is not successful none of the given subsequent instructions will be overridden.]

23. In regard to claim 42, Cocke discloses a data processing device reading in and executing instructions in a certain sequence, the data processing device comprising:

a. a fetch portion for reading in a computational instruction; [Figure 1 shows that instruction address are sent to an external storage or instruction memory for fetching as outlined in column 11, lines 65-69. Table III of column 4 shows that two conditional branch instructions (a first and second conditional instruction) are executed along with other operations (OP) and an Exit instruction. Column 1, line 73 – column 2, line 27 show that the conditions of the branches are determined or computed and thus the conditional branch instructions are computational instructions.]

b. a decoding portion for decoding the computational instruction that has been read in; [Figure 1A, element 40 illustrates an instruction decoder that decodes branch and other instructions as shown in column 2, lines 49-51 and thus the computational instruction is also decoded by this portion.]

c. an execution portion for executing the computational instruction in accordance with a decoding of the computational instruction; [Column 1, line 73 – column 2, line 27 show that instructions are processed (executed) including the branch (computational) instruction. Figure 1A, element 62 shows the function generator that processes or executes the instructions. This execution portion is also shown to receive its data from the instruction decoder and thus the execution is done in accordance with the decoding of the computational instruction.]

e. and an instruction overriding control circuit for overriding at least one instruction of subsequent instructions that follow the computational instruction in response to a conditional execution status updated by a sequencer in accordance with the decoding of the computational instruction, [In the example of figure 7 along with column 17, line 5-column 18, line 11, it is shown that Opα5 is overridden or inhibited from being decoded since it is already fetched into the buffer. Further, as shown in this section and figures 7A and 7B, OPα6 is overridden by never being fetched into a row of the buffer. This override is done due to the condition of the computational instruction (branch1) having been found to be successful. This section also describes the hardware of figures 1 (including the inhibiting flip-flop 126) used to override the subsequent instructions (and thus the hardware is a instruction overriding control circuit). This section along with figure 1A shows that the status of the condition ("successful" in the example of figures 7) is updated in branch state flip-flop 64. This is done by the result of the

conditional function from the function generator and since the hardware involved in figure 1A for updating the flip-flop determines branch conditions and has bearing on the sequence of executed instructions it may be appropriately named a sequencer. As shown in the sections cited above and in Figure 1A, the condition of the branch instruction is executed based on or in accordance with the decoding of the instruction from the instruction decoder 40.]

f. wherein the instruction overriding control circuit overrides the at least one instruction by allowing the fetch portion to skip reading in the at least one instruction. [As shown above with respect to OPα6.]

24. In regard to claim 43, Cocke discloses the data processing device according to claim 42, wherein the fetch portion includes a plurality of buffers to override the at least one instruction. [As shown in figure 1A, the fetch portion includes a plurality of buffers (Row 0 to Row N) that override the Opα6 instruction by not allowing it to enter the buffers.]

25. In regard to claim 44, Cocke discloses the data processing device according to claim 42, wherein the computational instruction and a first instruction and a second instruction are sequentially arranged as an instruction sequence, wherein the overriding control circuit overrides either the first instruction or the second instruction in response to the conditional execution status updated by the sequence. [As shown in the example of figure 7, a sequence of instructions includes the computational (branch1), EXIT (a first instruction sequentially after branch1), and Opα6 (a second instruction sequentially after branch1) instructions. As shown with respect to this example above, the second

instruction (Opα6) will be ignored or overridden in response to the conditional execution status of the first branch being updated by the sequencer to a successful state.]

Response to Arguments

26. Applicant's arguments filed 9/1/04 have been fully considered but they are not persuasive.

27. Applicant has argued that Cocke does not disclose or suggest an instruction overriding control circuit for controlling an overriding of subsequent instructions that follow a computational instruction in response to a conditional execution status updated by a sequencer in accordance with decoding of the computational instruction as recited by claim 25. As shown above in the rejection of claim 25: Column 1, line 73 – column 2, line 27 show that instructions are processed (executed) including the branch (computational) instruction. Column 4, lines 1-16 show that subsequent branches are ignored or overridden if the condition for a first branch condition is successful (conditional execution status). Column 17, lines 5-65 with further show this to be true with the example of figures 7 and 7A where the computational instruction is branch1. This section also describes the hardware of figures 1 (including the inhibiting AND gate) used to override the subsequent instructions (and thus the hardware is a instruction overriding control circuit). This section along with figure 1A shows that the status of the condition ("successful" in the example of figures 7) is updated in branch state flip-flop 64. This is done by the result of the conditional function from the function generator and since the hardware involved in figure 1A for updating the flip-flop determines branch conditions and has bearing on the sequence of executed instructions it may be

appropriately named a sequencer. As shown in the sections cited above and in Figure 1A, the condition of the branch instruction is executed based on or in accordance with the decoding of the instruction from the instruction decoder 40.

Conclusion

28. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

29. The following is text cited from 37 CFR 1.111(c): In amending in reply to a rejection of claims in an application or patent under reexamination, the applicant or patent owner must clearly point out the patentable novelty which he or she thinks the claims present in view of the state of the art disclosed by the references cited or the objections made. The applicant or patent owner must also show how the amendments avoid such references or objections.

30. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. The references cited in the previous Office Action remain pertinent and are cited herein by reference.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Shane F Gerstl whose telephone number is (571) 272-4166. The examiner can normally be reached on M-F 6:45-4:15 (First Friday Off).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (571) 272-4162. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Shane F Gerstl
Examiner
Art Unit 2183

SFG
November 17, 2004

Eddie Chan

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100